jh

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/800,109	03/12/2004	Christina Woody Mercier	07575-032002	8926

26181 7590 05/06/2005

FISH & RICHARDSON P.C.
PO BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
WASSUM, LUKE S	
ART UNIT	**PAPER NUMBER**
2167	

DATE MAILED: 05/06/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/800,109	Applicant(s) MERCIER ET AL.	
	Examiner Luke S. Wassum	Art Unit 2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 February 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *23-45* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *23-45* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *12 March 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *20040624*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

The Invention

1. The claimed invention is an apparatus providing coherent data copying operations where data replication is controlled by a source storage controller directly to a destination controller and managed by a remote application.

Priority

2. The examiner acknowledges the Applicants' claim to domestic priority under 35 U.S.C. § 120, as a continuation of application 09/375,819, filed 16 August 1999.

Information Disclosure Statement

3. The Applicants' Information Disclosure Statement, filed 24 June 2004, has been received and entered into the record. Since the Information Disclosure Statement complies with the provisions of MPEP § 609, the references cited therein have been considered by the examiner. See attached form PTO-1449.

Specification

4. The disclosure is objected to because of the following informalities:

For benefit claims under 35 U.S.C. 120, the reference to the parent application in the specification should include the current status of the application.

There is a typographical error on page 3, line 12: "...in the source objection...." should be "...in the source *object*...".

Appropriate correction is required.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 31-38 and 40-45 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

7. Claim 31 recites the limitation "the...write commands" in line 11. There is insufficient antecedent basis for this limitation in the claim.

8. Claims 32-38, fully incorporating the deficiencies of their parent claim, are likewise rejected.

9. Claim 40 recites the limitation "the copy logic" in the last limitation. There is insufficient antecedent basis for this limitation in the claim.

10. Claims 41-45, fully incorporating the deficiencies of their parent claim, are likewise rejected.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

13. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

14. Claims 23-26, 28, 29, 31-34, 36, 37, 39-41 and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Ohran et al.** (U.S. Patent 5,649,152) in view of **Hitz et al.** ("File System Design

for an NFS File Server Applicance") in view of **Schwartz et al.** ("LFS - A Local File System for Multiprocessor NFS Network Servers").

15. Regarding claim 23, **Ohran et al.** teaches a system substantially as claimed, comprising:

a) snapshot logic (see Abstract, disclosing that the reference is a method for providing a static snapshot; see also col. 1, lines 15-18);

b) copy logic (see disclosure that blocks are copied into the preservation memory when they are going to be changed by a write operation, col. 2, lines 55-58; see also col. 5, lines 48-53; see also step 212 in Figure 2);

c) an internal cache (see disclosure of block association memory, element 108 of Figure 1; see also col. 4, lines 51-56, disclosing that the block association memory may be a portion of the RAM of digital computer 102);

d) the system being operable to communicate with a replication manager to receive a snapshot command issued by the replication manager, the snapshot command specifying a range of data bytes of a source volume (see disclosure of a user indicating that a static image [i.e., snapshot] of the mass storage system is desired, said indication being analogous to the claimed snapshot command, col. 4, lines 14-24; note also the disclosure that mass storage system 104 can be any writable block-addressable storage system, such as one or more disks or a partition of a disk, a partition being a fixed portion of a disk, col. 3, lines 50-56; see also col. 5, lines 23-41);

e) the system being operable to communicate with the replication manager to receive a copy command specifying the source volume and target volume (see disclosure of the copy command, col. 5, lines 48-53);

f) the system being operable to receive a write command specifying the source volume (see disclosure of the intercepting of write commands to the source volume, col. 4, lines 35-41);

g) the snapshot logic being operable, in response to the snapshot command, to take a snapshot of the range, the snapshot including a snapshot map and snapshot data, the snapshot map being stored by the snapshot logic in the internal cache and the snapshot data being stored by the snapshot logic in a snapshot volume (see col. 4, lines 20-35; see also disclosure that preservation memory [i.e. the snapshot data] can be an area of memory, one or more disks, a partition of a disk, or a file stored on a disk, col. 3, line 66 through col. 4, line 1; see also disclosure that block association memory [i.e. the snapshot map] can be stored as a portion of the RAM of digital computer 102, col. 4, lines 51-56); and

h) the copy logic being operable in response to receiving the copy command to generate and send one or more storage device commands to one or more storage devices for the source and target volumes to copy data from the source volume to the target volume, the copy logic using the snapshot map and the snapshot data to maintain coherency of the copied data (see disclosure in the Abstract; see detailed disclosure of this process at col. 5, line 48 through col. 6, line 40).

Ohran et al. does not explicitly teach a system wherein the snapshot operations are carried out in and managed by a storage device controller.

Hitz et al., however, teaches a system wherein the snapshot operations are carried out in and managed by a storage device controller (see disclosure in both the Abstract (page 4) and the Introduction (page 5) that the system is a storage device controller managing snapshots).

It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate snapshot functionality in a storage device controller, since this would off-load the processing load for managing snapshots (as well as management of the file system itself) from the server to the storage device controller, thus improving performance on the server itself.

Neither **Ohran et al.** nor **Hitz et al.** explicitly teaches a storage device controller wherein the data is directly transferred between the source and destination storage device controllers without traversing a file server.

Schwartz et al., however, teaches teach a storage controller wherein the data is directly transferred between the source and destination storage device controllers without traversing a file server (see discussion of the LFS at section 1.2, pages 2 and 3; see also Figure 2, illustrating the fact that all network, file system and storage processing is completely removed from the Unix host processor and performed instead by dedicated processors; see also sections 4 and 5, pages 6-8).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transfer data directly from the source device controllers to the destination device controllers, since this would completely separate the Unix file system out of the kernel, thus resulting in a streamlined

tailored-for-NFS file system interface (see section 1.2, pages 2-3; see also sections 4 and 5, pages 6-8).

16. Regarding claim 31, **Ohran et al.** teaches a method substantially as claimed, comprising:

a) receiving a snapshot command issued by a replication manager, the snapshot command specifying a range of data bytes of a source volume (see disclosure of a user indicating that a static image [i.e., snapshot] of the mass storage system is desired, said indication being analogous to the claimed snapshot command, col. 4, lines 14-24; note also the disclosure that mass storage system 104 can be any writable block-addressable storage system, such as one or more disks or a partition of a disk, a partition being a fixed portion of a disk, col. 3, lines 50-56; see also col. 5, lines 23-41);

b) in response to receiving the snapshot command, the system taking a snapshot of the range specified using device control commands to control one or more devices on which the source is stored, the snapshot including a snapshot map and snapshot data, and storing the snapshot map and the snapshot data in a cache internal to the system and a snapshot volume, respectively (see col. 4, lines 20-35; see also disclosure that preservation memory [i.e. the snapshot data] can be an area of memory, one or more disks, a partition of a disk, or a file stored on a disk, col. 3, line 66 through col. 4, line 1; see also disclosure that block association memory [i.e. the snapshot map] can be stored as a portion of the RAM of digital computer 102, col. 4, lines 51-56);

c) receiving a copy command from the replication manager, the copy command specifying a copy operation from the source volume to a target volume (see disclosure of the copy command, col. 5, lines 48-53); and

d) in response to receiving the copy and write commands, the system generating and sending storage device commands to one or more storage devices of the source and target volumes to copy data from the source volume to the target volume using the snapshot map and snapshot data to maintain coherency of the copied data (see disclosure in the Abstract; see detailed disclosure of this process at col. 5, line 48 through col. 6, line 40).

Ohran et al. does not explicitly teach a method wherein the snapshot operations are carried out in and managed by a storage device controller.

Hitz et al., however, teaches a method wherein the snapshot operations are carried out in and managed by a storage device controller (see disclosure in both the Abstract (page 4) and the Introduction (page 5) that the system is a storage device controller managing snapshots).

It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate snapshot functionality in a storage device controller, since this would off-load the processing load for managing snapshots (as well as management of the file system itself) from the server to the storage device controller, thus improving performance on the server itself.

Neither **Ohran et al.** nor **Hitz et al.** explicitly teaches a storage device controller wherein the data is directly transferred between the source and destination storage device controllers without traversing a file server.

Schwartz et al., however, teaches teach a storage controller wherein the data is directly transferred between the source and destination storage device controllers without traversing a file server (see discussion of the LFS at section 1.2, pages 2 and 3; see also Figure 2, illustrating the fact that all network, file system and storage processing is completely removed from the Unix host processor and performed instead by dedicated processors; see also sections 4 and 5, pages 6-8).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transfer data directly from the source device controllers to the destination device controllers, since this would completely separate the Unix file system out of the kernel, thus resulting in a streamlined tailored-for-NFS file system interface (see section 1.2, pages 2-3; see also sections 4 and 5, pages 6-8).

17. Regarding claim 39, **Ohran et al.** teaches a computer-implemented method substantially as claimed, comprising:

a) using a remote application to manage a source storage device controller and a destination storage device controller (see col. 3, lines 43-59);

b) generating a snapshot version for each block of the source data object changed by one or more write operations to the block during the course of a copy operation (see

disclosure in the Abstract; see detailed disclosure of this process at col. 5, line 48 through col. 6, line 40); and

c) copying each block of the source data object to a corresponding block in the destination data object in the absence of the snapshot version of the block and otherwise copying the snapshot version of the source data object block to the corresponding block in the destination data object, wherein data is transferred between the source and destination storage device controllers (see disclosure in the Abstract; see detailed disclosure of this process at col. 5, line 48 through col. 6, line 40).

Ohran et al. does not explicitly teach a method wherein the snapshot operations are carried out in and managed by a storage device controller, thus maintaining coherency without requiring any file system to maintain a snapshot map.

Hitz et al., however, teaches a method wherein the snapshot operations are carried out in and managed by a storage device controller, thus maintaining coherency without requiring any file system to maintain a snapshot map (see disclosure in both the Abstract (page 4) and the Introduction (page 5) that the system is a storage device controller managing snapshots).

It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate snapshot functionality in a storage device controller, since this would off-load the processing load for managing snapshots (as well as management of the file system itself) from the server to the storage device controller, thus improving performance on the server itself.

Neither **Ohran et al.** nor **Hitz et al.** explicitly teaches a storage device controller wherein the data is directly transferred between the source and destination storage device controllers without traversing a file server.

Schwartz et al., however, teaches teach a storage controller wherein the data is directly transferred between the source and destination storage device controllers without traversing a file server (see discussion of the LFS at section 1.2, pages 2 and 3; see also Figure 2, illustrating the fact that all network, file system and storage processing is completely removed from the Unix host processor and performed instead by dedicated processors; see also sections 4 and 5, pages 6-8).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transfer data directly from the source device controllers to the destination device controllers, since this would completely separate the Unix file system out of the kernel, thus resulting in a streamlined tailored-for-NFS file system interface (see section 1.2, pages 2-3; see also sections 4 and 5, pages 6-8).

18. Regarding claim 40, **Ohran et al.** teaches a system substantially as claimed, comprising:

a) a replication manager that is operable to issue a snapshot command (see disclosure of a user indicating that a static image [i.e., snapshot] of the mass storage system is desired, said indication being analogous to the claimed snapshot command, col. 4, lines 14-24; note also the disclosure that mass storage system 104 can be any writable block-

addressable storage system, such as one or more disks or a partition of a disk, a partition being a fixed portion of a disk, col. 3, lines 50-56; see also col. 5, lines 23-41);

b) a storage device controller that is operable to:

i) communicate with the replication manager to receive the snapshot command specifying a range data bytes of the source volume (see disclosure of a user indicating that a static image [i.e., snapshot] of the mass storage system is desired, said indication being analogous to the claimed snapshot command, col. 4, lines 14-24; note also the disclosure that mass storage system 104 can be any writable block-addressable storage system, such as one or more disks or a partition of a disk, a partition being a fixed portion of a disk, col. 3, lines 50-56; see also col. 5, lines 23-41); and

ii) receive a copy command specifying the source volume and target volume (see disclosure of the copy command, col. 5, lines 48-53); wherein

c) the controller is operable to receive a write command specifying the source volume (see disclosure of the intercepting of write commands to the source volume, col. 4, lines 35-41);

d) the controller is operable, in response to receiving the snapshot command, to take a snapshot of the range, the snapshot including a snapshot map and snapshot data, the snapshot map being stored in a cache and the snapshot data being stored in a snapshot volume (see col. 4, lines 20-35; see also disclosure that preservation memory [i.e. the snapshot data] can be an area of memory, one or more disks, a partition of a disk, or a file stored on a disk, col. 3, line 66 through col. 4, line 1; see also disclosure

that block association memory [i.e. the snapshot map] can be stored as a portion of the RAM of digital computer 102, col. 4, lines 51-56); and

e) the controller is operable, in response to receiving the copy command, to generate and send one or more storage device commands to one or more storage devices for the source and target volumes to copy data from the source volume to the target volume, the copy logic using the snapshot map and the snapshot data to maintain coherency of the copied data (see disclosure in the Abstract; see detailed disclosure of this process at col. 5, line 48 through col. 6, line 40).

Ohran et al. does not explicitly teach a system wherein the snapshot operations are carried out in and managed by a storage device controller.

Hitz et al., however, teaches a system wherein the snapshot operations are carried out in and managed by a storage device controller (see disclosure in both the Abstract (page 4) and the Introduction (page 5) that the system is a storage device controller managing snapshots).

It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate snapshot functionality in a storage device controller, since this would off-load the processing load for managing snapshots (as well as management of the file system itself) from the server to the storage device controller, thus improving performance on the server itself.

Neither **Ohran et al.** nor **Hitz et al.** explicitly teaches a storage device controller wherein the data is directly transferred between the source and destination storage device controllers without traversing a file server.

Schwartz et al., however, teaches teach a storage controller wherein the data is directly transferred between the source and destination storage device controllers without traversing a file server (see discussion of the LFS at section 1.2, pages 2 and 3; see also Figure 2, illustrating the fact that all network, file system and storage processing is completely removed from the Unix host processor and performed instead by dedicated processors; see also sections 4 and 5, pages 6-8).

It would have been obvious to one of ordinary skill in the art at the time of the invention to transfer data directly from the source device controllers to the destination device controllers, since this would completely separate the Unix file system out of the kernel, thus resulting in a streamlined tailored-for-NFS file system interface (see section 1.2, pages 2-3; see also sections 4 and 5, pages 6-8).

19. Regarding claims 24 and 32, **Hitz et al.** additionally teaches a storage device controller and method wherein the storage device controller is a RAID controller (see section 1.0 <u>Introduction</u>, page 5, third paragraph).

20. Regarding claims 25 and 33, **Ohran et al.** additionally teaches a storage device controller and method wherein:

a) the range of the storage volume specified by the snapshot command is a first range, and the write command specifies a second range of data bytes of the source volume (see disclosure of a user indicating that a static image [i.e., snapshot] of the mass storage system is desired, said indication being analogous to the claimed snapshot command, col. 4, lines 14-24; note also the disclosure that mass storage system 104 can be any writable block-addressable storage system, such as one or more disks or a partition of a disk, a partition being a fixed portion of a disk, col. 3, lines 50-56; see also col. 5, lines 23-41; see also disclosure of the intercepting of write commands to the source volume, col. 4, lines 35-41); and

b) the controller is operable, in response to receiving the write command while the source volume is being copied to the target volume, to hold the write command in the cache, check if the first range overlaps with the second range and, if so, copy the second range from the source volume to the snapshot volume, update the snapshot map, and then allow the write command to write to the source volume (see disclosure in the Abstract; see detailed disclosure of this process at col. 5, line 48 through col. 6, line 40; see also flowchart illustrated in Figure 2).

21. Regarding claims 26 and 34, **Ohran et al.** additionally teaches a storage device controller and method wherein the replication manager is executed on a file server (see col. 6, lines 50-55).

22. Regarding claims 28, 36 and 41, **Ohran et al.** additionally teaches a storage device controller, system and method wherein the replication manager is operable to control multiple storage device controllers (see col. 6, lines 40-49).

23. Regarding claims 29 and 37, **Ohran et al.** additionally teaches a storage device controller and method wherein the one or more storage device commands include SCSI commands (see disclosure that the system includes a mass storage device that could be a SCSI device, col. 3, lines 60-65).

24. Regarding claim 45, **Ohran et al.** additionally teaches a storage device controller wherein a block size is specified so that fixed size blocks are written to the destination controller device (see col. 5, lines 23-41).

25. Claims 27 and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Ohran et al.** (U.S. Patent 5,649,152) in view of **Hitz et al.** ("File System Design for an NFS File Server Applicance") in view of **Schwartz et al.** ("LFS - A Local File System for Multiprocessor NFS Network Servers") as applied to claims 23-26, 28, 29, 31-34, 36, 37, 39-41 and 45 above, and further in view of **Tawil** (U.S. Patent 6,421,723).

26. Regarding claims 27 and 35, **Ohran et al.**, **Hitz et al.** and **Schwartz et al.** teach a storage device controller and method substantially as claimed.

None of **Ohran et al.**, **Hitz et al.** nor **Schwartz et al.** explicitly teaches a storage device controller and method wherein the file server is connected to a storage area network switch and the file server communicates with the storage device controller through the storage area network switch.

Tawil, however, teaches the use of a storage area network (see col. 1, lines 30-42).

It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate a storage area network, since they offer centralized storage of data for increased efficiency and data handling, and provide data access reliability and availability, unobtrusive capacity expansion, improved data backup and recovery, and performance that is competitive with local data storage (see col. 1, lines 30-36).

27. Claims 30 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Ohran et al.** (U.S. Patent 5,649,152) in view of **Hitz et al.** ("File System Design for an NFS File Server Applicance") in view of **Schwartz et al.** ("LFS – A Local File System for Multiprocessor NFS Network Servers") as applied to claims 23-26, 28, 29, 31-34, 36, 37, 39-41 and 45 above, and further in view of **Dulai et al.** (U.S. Patent 6,205,479).

28. Regarding claims 30 and 38, **Ohran et al.**, **Hitz et al.** and **Schwartz et al.** teach a storage device controller and method substantially as claimed.

None of **Ohran et al.**, **Hitz et al.** nor **Schwartz et al.** explicitly teaches a storage device controller and method wherein the controller is operable to send the one or more storage device commands by using one of an in-band protocol or an out-of-band protocol.

Dulai et al., however, teaches a storage device controller and method wherein the controller is operable to send the one or more storage device commands by using one of an in-band protocol or an out-of-band protocol (see disclosure of the use of an in-band protocol, claims 18 and 21).

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize an in-band protocol, since this allows the transmission of commands over a widely dispersed network where the use of an out-of-band protocol might be impractical.

29. Claims 42-44 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Ohran et al.** (U.S. Patent 5,649,152) in view of **Hitz et al.** ("File System Design for an NFS File Server Applicance") in view of **Schwartz et al.** ("LFS – A Local File System for Multiprocessor NFS Network Servers") as applied to claims 23-26, 28, 29, 31-34, 36, 37, 39-41 and 45 above, and further in view of **Simpson et al.** (U.S. Patent 6,128,306).

30. Regarding claims 42-44, **Ohran et al.**, **Hitz et al.** and **Schwartz et al.** teach a storage device controller and method substantially as claimed.

None of **Ohran et al.**, **Hitz et al.** nor **Schwartz et al.** explicitly teaches a storage device controller and method comprising a list of blocks to be copied which is reordered to optimize copy speed, wherein control data is inserted before and after the source data block, nor wherein the list is buffered.

Simpson et al., however, teaches a storage device controller and method comprising a list of blocks to be copied which is reordered to optimize copy speed (see col. 2, lines 15-18), wherein control data is inserted before and after the source data block (see col. 2, lines 5-9), and wherein the list is buffered (see col. 1, lines 55-58).

It would have been obvious to one of ordinary skill in the art at the time of the invention to include prioritized buffering of output data, since this allows more flexible handling of outgoing data traffic, and furthermore since input/output buffering and prioritization and reordering of data in queues was well known in the art at the time of the invention.

Conclusion

31. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Blea et al. (U.S. Patent 6,212,531) teaches a method for performing a point-in-time backup using a snapshot function.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Luke S. Wassum whose telephone number is 571-272-4119. The examiner can normally be reached on Monday-Friday 8:30-5:30, alternate Fridays off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John E. Breene can be reached on 571-272-4107. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

In addition, INFORMAL or DRAFT communications may be faxed directly to the examiner at 571-273-4119.

Customer Service for Tech Center 2100 can be reached during regular business hours at (571) 272-2100, or fax (703) 872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Luke S Wassum
Luke S. Wassum
Primary Examiner
Art Unit 2167

lsw
2 May 2005